In the Specification:

Please amend the Specification as follows:

On page 3, fourth full paragraph, amend line 10 to read as follows:

FIG. 2 illustrates a high level diagram of a system having a combined voice and data network;

On page 3, ninth full paragraph, amend line 16 to read as follows:

FIG. 7 illustrates a more detailed diagram of the system of FIG. 6;

On page 3, eleventh full paragraph, amend lines 18-19 to read as follows:

FIG. [[12]] 9 illustrates the contents of a message sent over the network based on active channel reordering to deallocate channel groups;

On page 3, twelfth full paragraph, amend lines 20-21 to read as follows:

FIG. [[13]] 10 illustrates a PBX telephone system comprised of PBX main and expansion cabinets connected via a data network;

On page 3, thirteenth full paragraph, amend line 22 to read as follows:

FIG. [[14]] 11 illustrates the transmission of voice packets in a main or expansion cabinet;

On page 3, fourteenth full paragraph, amend line 23 to read as follows:

FIG. [[15]] 12 illustrates the reception of voice packets in a main or expansion cabinet;

On page 3, fifteenth full paragraph, amend line 24 to read as follows:

FIG. [[16a]] 13a illustrates operation of a packet loss counter;

On page 3, sixteenth full paragraph, amend line 25 to read as follows:

FIG. [[16b]] 13b further illustrates operation of a packet loss counter;

On page 3, seventeenth full paragraph, amend line 26 to read as follows:

FIG. [[17]] 14 is a diagram showing operation of packet round trip time calculation;

On page 3, eighteenth full paragraph, amend line 27 to read as follows:

FIGS. [[18a]] 15a-[[18b]] 15b illustrate operation of bandwidth optimization using idle card elimination[[.]];

On page 3, nineteenth full paragraph, amend lines 28-29 to read as follows:

FIGS. [[19a]] 16a, [[19b]] 16b and [[19c]] 16c illustrate operation of bandwidth optimization using priority-based card elimination;

On page 3, twentieth full paragraph, amend line 30 to read as follows:

FIG. [[20]] 17 is a flow chart detailing operation of a packet loss counter and system reaction;

On page 3, twenty-first full paragraph, amend lines 31-32 to read as follows:

FIG. [[21]] 18 is a flow chart showing operation of packet delay variation measurement and system reaction; and

On page 4, first full paragraph, amend line 1 to read as follows:

FIG. [[22]] 19 is a flow chart diagram showing implementation of bandwidth optimization.

On page 7, first full paragraph, amend lines 3-14 to read as follows:

Remote CPU 750, switching matrix 760, and master CPU 790 could be a number of machines, a separate machine, or a portion of a machine. For example, as shown in FIG. 7, each of remote CPU 750, switching matrix 760, and master CPU 790 could reside in cabinets that communicate via data link 710. For example, each of cabinets 800, 840, and 880 includes a memory 801, 841, and 881; secondary storage 802, 842, and 882; a central processing unit

(CPU) ~~790~~ 803, 843, and 883; an input device 804, 844, and 884; a video display 805, 845, and 885; and slots 806, 846, and 886. One skilled in the art will appreciate that cabinets 800, 840, and 880 may contain additional or different components and that each cabinet could include the same hardware as the other cabinets or different hardware. Each of memories 801, 841, and 881 includes an operating system 807, 847, and 887; a TCP/IP protocol stack 808, 848, and 888; an active communication detection program 809, 849, and 889; a table management program 810, 850, and 890; and a communication program 811, 851, and 891.

On page 7, third full paragraph, amend lines 26-32 to read as follows:

As shown in FIG. 8, each of switching matrixes ~~60~~ 760, 855, and 895 include a multiplexer/demultiplexer 900 and a register 910. Although register 910 is shown as separate from multiplexer/demultiplexer 900, multiplexer/demultiplexer 900 and register 910 could be combined in a single device. Multiplexer/demultiplexer 900 formats and receives packets sent over data link 710 and outputs parallel data to a switch 920 or an IP packet to data link 710, using, for example, a field programmable gate array. Register 910 stores a value that indicates the maximum channel number in the packet.

On page 8, sixth full paragraph, amend lines 23-27 to read as follows:

Referring now to FIG. [[13]] 10, an objective of the aforementioned development is to provide the capability to support communication (including voice) between two or more PBXs over a data network, such as an IP LAN 10. In one configuration, one PBX 12 acts as a master (also referred to as the main cabinet) containing a master CPU and master switching matrix and the other PBXs 14 are slaved (also referred to as the expansion cabinets).

On page 9, fourth full paragraph, amend lines 20-25 to read as follows:

FIG. [[14]] 11 illustrates the connections for transmission of voice packets. PBX cabinet 20, which can be either a main or expansion cabinet, is typically able to accommodate up to 320 voice channels. (The maximum number of channels can be configured by the user or the software to include a lesser or greater number of channels.) Daughter board 22 incorporating a

FPGA IC is connected to cabinet 20. Each transmitted packet 24 typically contains a maximum of 320 PCM samples. The transmitted packet 26 is forwarded to the network via IP port 28.

On page 9, fifth full paragraph, amend line2 26-31 to read as follows:

Similarly, PBX cabinet 30, shown in FIG. [[15]] 12, which can be either a main or expansion cabinet incorporates an IP port 38 which receives incoming packet 36. Typically, a FPGA incorporated within board 32 is capable of receiving packets containing, for example, up to 320 PCM samples/channels. The received packet 34 populated with the samples is processed by the FPGA IC so that each byte of the voice frame is re-mapped to unique channels, one for each of the bytes, shown as channels 1, 2...320 in cabinet 30.

Beginning on page 11, fifth full paragraph continuing on page 12, amend line 23 on page 11 through line 2 on page 12 to read as follows:

Packet Loss Counter (PLC). One way of implementing a Packet Loss Counter may be via a hardware register, as shown in FIG. [[16a]] 13a. Each voice packet is labelled (within its payload) with a sequence number. The receiving end monitors the sequence of packets and as each packet is received when a break in sequence of the arriving packets occurs, the counter is incremented by one digit. A number of packets 47a-c, 49a-c, 51a-c forming data streams 46, 48, 50 are shown entering a plurality of IP ports 41, 43, 45, respectively. When a packet arrives out of sequence, e.g., a sequence such as {1, 3,} (corresponding to packets 47a, 47b, 47c), the counter is incremented by 1. A sequence such as {1, 6, 7} (corresponding to packets 49a, 49b, 49c) results in incrementing the counter by 1, since packet numbers 2, 3, 4 and 5 are missing from the data stream. A sequence such as {1, 3, 2} (corresponding to packets 51a, 51b, 51c) will result in the counter being incremented by 2, since packet #2 arrives out of sequence (after packet #3). Thus, the value on the counter is an indicator of the degree of packet loss.

On page 12, first full paragraph, amend lines 3-11 to read as follows:

FIG. [[16b]] 13b depicts an alternative hardware implementation of a Packet Loss Counter implementation showing data stream 52, including representative packets 53a, 53b and 53c (numbered #1, #2...#7000). Using the total number of packets expected to arrive per

second (e.g., 8000) as a reference, when each packet arrives (a counter will be incremented until the end of time period (second) here shown as 7000), the balance of 1000 comprising the packets that did not arrive at the port within the given time frame represents the number of packets lost. This measurement may be done at short intervals (e.g., every second). The counter is then reset to the reference number (e.g. 8000). In FIG. [[16b]] 13b which shows the packets traveling as a function of time, the packet loss counter will therefor read 1000 after the first second.

On page 12, third full paragraph, amend lines 18-25 to read as follows:

Latency. Latency is also an indicator that the network has become congested. One way of recording the round trip time required for voice packets to transit a network is to use a hardware register. Referring to FIG. [[17]] 14, source 500 marks packet 502 as #1 (Step 1) prior to sending it and also starts timer 504. When the destination machine 506 receives packet 502 (Step 2), it complements the value that will be transmitted on the next packet as packet #2 (508) back to the source (Step 3). Upon the arrival of the #2-marked packet (Step 4), the timer stops and the time difference is stored in the trip register. Such marking is done within the payload of the respective voice packets.

On page 13, first full paragraph, amend lines 4-8 to read as follows:

Bandwidth. A software module provides measurement of bandwidth. One way of performing this measurement is to sum the total length of arrived packets per second minus the number of packets lost (obtained from the packet loss counter). All the transmitted and received packets are periodic. Another alternative is to use TCP/UDP/ICMP to record the round trip time to generate packets of different sizes and to record their round trip time.

On page 14, fifth full paragraph, amend lines 27-28 to read as follows:

Referring now to FIG. [[18a]] 15a, there is shown a technique for Idle Card Elimination according to a feature of the invention.

Beginning on page 14, sixth full paragraph continuing on page 15, amend line 29 on page 14 through line 8 on page 15 to read as follows:

A packet 180, here shown as comprising three cards, 186, 188, 190, also includes IP header 182 and a UDP header 184. Each card has 32 channels assigned to it. Certain channels are active (designated in the Figure as "A") and others are inactive or idle (designated as "I"). In the example shown, all of the channels in the second card 188 are idle. With the Idle Card Elimination feature as shown in FIG. [[18b]] 15b, the channels associated with the second card are eliminated, the channels associated with the third card, 190, are mapped as channels associated with card #2 and the system recognizes that packet 1800, containing header information 1820, 1840 now contains two cards, 1860 and 1880, each having at least some channels active. The bandwidth is thus optimized by reducing the number of channels associated with cards in the transmitted packet. This can be achieved by searching the switching matrix status table described earlier for a group of channels collocated with each other with [[an]] a starting index matching the index of the first channel on a given card.

On page 15, sixth full paragraph, amend lines 25-26 to read as follows:

Referring now to FIG. [[19a]] 16a, there is shown a technique for card priority assignment according to a feature of the invention.

Beginning on page 15, seventh full paragraph continuing on page 16, amend line 27 on page 15 through line 3 on page 16 to read as follows:

A packet 190, here shown as comprising three cards, 196, 198, 200, also includes an IP header 192 and a UDP header 194. Each card has 32 channels assigned to it. In this example, the system is configured to assign the highest priority to the second card 198, and the lowest priority to the first card, 196. When network conditions are such that bandwidth optimization needs to be enabled, the system drops the card having the lowest priority. In this particular example, channels associated with card 196 (having the lowest priority) are dropped and channels associated with cards 198 and 200 are shifted in position. As shown in FIG. [[19b]] 16b, channels associated with card 198 are then remapped as the first card, 1960, and channels

associated with the third card are remapped as if they were associated with the second card, 2000.

On page 16, first full paragraph, amend lines 4-9 to read as follows:

This process may be continued based on the next lower priority of the remaining cards. Thus, the channel associated with the third card 2000 in FIG. [[19b]] 16b was, for example, assigned the lowest priority and the channel associated with the second card 1960, the highest. FIG. [[19c]] 16c shows the resulting configuration after implementing priority based bandwidth optimization. The channels associated with a single card 1904 (corresponding to card 1960), 198, remains in the packet. Again, the number of channels associated with cards in the transmitted packet has been reduced.

On page 17, fifth full paragraph, amend line 25-28 to read as follows:

Provided the second PLC hardware implementation is used (i.e. measuring the number of packets received, as shown in FIG. [[16b]] 13b), the measurement will be performed every second. Consequently, $\Delta_t = 1$. If the first PLC implementation is used, either a fixed or changing time window can be used to read the hardware register for obtaining the measurement.

On page 24, first full paragraph, amend lines 1-12 to read as follows:

Referring to FIG. [[20]] 17, there is shown a flow chart depicting a process for measuring the amount of packet loss as an adjunct to enabling bandwidth optimization. A complete packet interval needed to fill the Packet Loss Counter hardware register described earlier is awaited (step 60). Next, the Packet Loss Counter (PLC) is read (step 62). If the PLC reading exceeds a predetermined threshold (expressed as PLC > Thr1) and, if the current PLC reading compared to the previous PLC reading is greater than zero (Δ PLC > 0) (step 64), the PLC flag is incremented by 1 (step 66). Otherwise, the process of waiting for a complete packet interval is repeated (step 60). After the PLC flag is incremented (step 66), the PLC flag reading is compared to a second predetermined threshold (PLC_Flag>thr2, step 68). If the value of the flag exceeds that of the threshold, the network is inferred congested and the program proceeds

to enable bandwidth optimization (step 70). Enablement also causes the PLC flag to be reset to zero (step 72) at which point the system ~~and~~ awaits a new complete packet interval (step 60).

On page 24, second full paragraph, amend lines 13-16 to read as follows:

FIG. [[21]] 18 shows a flow chart for measuring packet delay variation. As previously mentioned, packets arriving faster than the current system is set up to process them can result in loss of data. Conversely, packets arriving too slowly may result in gaps in the data, noticeable as pauses during voice conversations.

On page 24, third full paragraph, amend lines 17-22 to read as follows:

A complete packet interval needed to fill the Packet Delay Variation (PDV) hardware buffer, described earlier, is awaited (step ~~70~~ 71). The number of events of overflow and underflow in the PDV buffer is then read (step ~~72~~ 73) and if there is any overflow (OverFlow>0, step 74), value of the overflow variable is incremented (step 76). The PDV_OF value is then compared with a predetermined overflow threshold value (PDV_OF>OF_Thr, step 78). If the PDV_OF value exceeds that of OF_Thr, the PDV_OF variable is reset to zero (step 82).

On page 24, fourth full paragraph, amend lines 23-24 to read as follows:

The PDV overflow threshold has not been exceeded (step 78), a new packet interval time slot is awaited (step ~~70~~ 71) and the steps are repeated.

On page 25, first full paragraph, amend lines 1-9 to read as follows:

With continuing reference to FIG. [[21]] 18, no overflow is detected (step 74) the packet underflow count is compared to zero (step 84). If the count does not exceed zero, a new packet interval branches back to step ~~70~~ 71 and a time slot is awaited. On the other hand, if the underflow count is greater than zero, the underflow variable PDF_UF is incremented (step 86). Program step 88 is a logic step which compares the underflow variable from step 86 to a predetermined underflow threshold (UF_Thr). If PDV_UF is greater than UF_Thr, the buffer size is reduced at step 90 and the PDV_UF variable is reset to zero in step 92. If PDV_UF is

less than UF_Thr, at logical step 88, the program branches back to step ~~70~~ 71 to await the next packet interval. Upon resetting in step 92, the program also branches back to step ~~70~~ 71.

On page 25, second full paragraph, amend lines 10-12 to read as follows:

FIG. [[22]] 19 illustrates a flow chart showing implementation of bandwidth optimization. An indication from the program that bandwidth optimization is to be implemented, such as the enable bandwidth optimization program command 70 shown in FIG. [[20]] 17 initiates the process.